

REMARKS

This communication is in response to the Final Office Action mailed on July 9, 2008 in which claims 1, 5-16, and 19-28 were pending. With this response, claim 8 is amended. Applicant respectfully requests entry of the amendment to claim 8, because the resulting change does not necessitate an additional search, as it removes a portion of one element of the claim such that the resulting claim 8 is similar to other claims already under examination. With this response, all of the pending claims are in condition for allowance. Reconsideration and notice to that effect is respectfully requested.

Applicant respectfully traverses the rejection of claims 1, 5-6, 14-16 and 21-24 under 35 U.S.C. §103(a) over U.S. Patent No. 5,974,253 (“Nahaboo”) and U.S. Patent No. 5,493,508 (“Dangelo”). The asserted combination of Nahaboo and Dangelo fails to disclose or suggest all of the elements of the claims.

In general, the Office blurs a distinction between the interface development tool of Nahaboo and the associated application with the embedded WOOL interpreter program that actually implements the developed (designed) user interface file. By ignoring that distinction, the Office then makes the extension that the interface development tool could be incorporated into the circuit design tool of Dangelo.

The asserted combination of Nahaboo and Dangelo fails to disclose or suggest “a command interpreter” that “modifies the graphical user interface at run time of the graphical user interface according to the interpreted configuration commands” including “building graphical objects within the graphical interface according to the interpreted configuration commands,” “assigning functionality to the built graphical objects according to the interpreted configuration commands,” and “displaying the graphical objects within the graphical interface according to the interpreted configuration commands,” “wherein the graphical objects are selectable by a user to access the assigned functionality to produce an integrated circuit design”, as recited in claim 1.

Nahaboo discloses a user interface description tool that uses an interpreted language where both data and the program have similar representations and that uses libraries of command objects

and graphical objects. *See Nahaboo*, Abstract. Nahaboo notes that poor interface designs can result in mitigated commercial success. *See Nahaboo*, col. 1, lines 25-26.

Nahaboo discloses that “the purpose of this invention is to define an extremely flexible interface development tool that can be used regardless of the application.” *See Nahaboo*, col. 1, lines 29-34 (emphasis added). Nahaboo discloses that the interface development tool uses a list processing language (LISP) type of interpreted language that can be interpreted by an interpreter (WOOL) language program. *See Nahaboo*, col. 1, lines 35-39. Further, Nahaboo discloses that the interface development tool produces an interactive assembly geometry specification that is translated to a file that can be used by an application and stored on a disk, and that the file contains the user-interface format in the form of a WOOL language program. *See Nahaboo*, col. 2, lines 10-14.

Nahaboo states that the interactive interface tool allows the user to create and modify interfaces and windows that exhibit behaviors that can be easily duplicated. *See Nahaboo*, col. 4, lines 39-60. However, Nahaboo discloses an interface development tool to design and refine interfaces that can be used with applications, but does not suggest that the user of the applications allow for design and refinement of the interface “at run time of the graphical user interface,” as recited in claim 1. Instead, Nahaboo discloses that the graphical user interface can be designed and edited and then stored for use with an application. At runtime of the application, the saved interface is loaded and executed using a WOOL program language interpreter that is embedded with the application. *See Nahaboo*, col. 1, lines 35-39. Thus, the WOOL interpreter program, **not the interface development tool**, is embedded with the application. While Nahaboo discloses that the interface description program can be expanded by adding other “widget” classes, Nahaboo discloses that “these classes can be added by describing the new attribute types of each class in the WOOL language so that the new “widgets” can be edited, for example, using the menu of the interface development tool. *See Nahaboo*, col. 2, lines 21-56. In Nahaboo, since the WOOL interpreter program loads and interprets the WOOL language program, such modifications or additions would take effect only upon restarting the application. Thus, Nahaboo discloses modification of the graphical user interface using an interface development tool, storage of the

modified graphical user interface, and later execution using a WOOL interpreter program that is embedded with the application and that executes the stored graphical user interface file in conjunction with the application.

Applicant notes that Nahaboo discloses editing an interface using a separate application from that with which the interface is to be used. The edited interface is stored in a WOOL language program and the WOOL language interpreter is embedded with the application to interpret the interface file. However, Nahaboo does not disclose or suggest that the WOOL language interpreter “modifies the graphical user interface at run time of the graphical user interface”, as recited in claim 1. Accordingly, Nahaboo does not disclose or suggest “a command interpreter” that “modifies the graphical user interface at run time of the graphical user interface”, as recited in claim 1.

While Dangelo discloses that the schematic editors allow for selection of previously created and stored schematics, Dangelo does not disclose or suggest “a command interpreter” that “modifies the graphical user interface at run time of the graphical user interface”, as recited in claim 1.

Accordingly, the asserted combination of Nahaboo and Dangelo fails to disclose or suggest all of the elements of claim 1, or of claims 5-7, at least by virtue of their dependency from allowable claim 1. Therefore, the rejection of claims 1 and 5-7 over the asserted combination of Nahaboo and Dangelo should be withdrawn.

With respect to claim 8, the asserted combination of Nahaboo and Dangelo fails to disclose or suggest “upon execution of a command processor by a user, loading a top level Tool Command Language (TCL) command into a namespace, the command processor including a graphical user interface (GUI) without graphical objects,” “loading one or more TCL commands into the command processor from a command line in which the one or more TCL configuration commands are entered by the user,” “building graphical objects according to the TCL configuration commands,” “assigning functionality to the built graphical objects according to the TCL configuration commands”, and “displaying the graphical objects within the GUI according to the TCL configuration commands, the graphical objects selectable by the user to produce an integrated

circuit design”, as recited in claim 8.

As discussed above, Nahaboo discloses that the interface development tool allows for modification and design of interfaces, which are stored in a WOOL language program together with a WOOL interpreter program embedded within the application. *See Nahaboo*, col. 1, lines 35-39. The interface development tool of Nahaboo is not also used to produce an integrated circuit design. Rather, a separate application is executed that uses the GUI designed by the interface development tool of Nahaboo to display user selectable object. Nahaboo provides no suggestion or motivation for combining the interface development tool with the actual application that is executed by the user (and not the designer). Further, Nahaboo provides no suggest or teaching that the “loading one or more TCL commands into the command processor from a command line in which the one or more TCL configuration commands are entered by the user,” “building graphical objects according to the TCL configuration commands,” “assigning functionality to the built graphical objects according to the TCL configuration commands”, and “displaying the graphical objects within the GUI according to the TCL configuration commands, the graphical objects selectable by the user to produce an integrated circuit design”, as recited in claim 8. Thus, Nahaboo does not disclose all of the elements of claim 8.

Dangelo discloses that the previously designed circuit modules can be reused, but does not overcome the deficiencies of Nahaboo. In particular, Dangelo does not disclose a “command processor including a graphical user interface (GUI) without graphical objects,” and does not disclose “loading one or more TCL commands into the command processor from a command line in which the one or more TCL configuration commands are entered by the user,” “building graphical objects according to the TCL configuration commands,” “assigning functionality to the built graphical objects according to the TCL configuration commands”, and “displaying the graphical objects within the GUI according to the TCL configuration commands, the graphical objects selectable by the user to produce an integrated circuit design”, as recited in claim 8.

Accordingly, the asserted combination of Nahaboo and Dangelo is missing at least one element of claim 8, and of claims 9-16, at least by virtue of their dependency from allowable claim

8.

The asserted combination of Nahaboo and Dangelo fails to disclose or suggest “loading a configuration command of the one or more configuration commands into the command processor from at least one of a user specified command configuration script comprising the configuration command or from a command line in which the configuration command is entered by the user” and “assembling a graphical user interface having no hard coded objects and including at least one graphical user interface (GUI) component based on interpreted configuration commands from the user, the at least one graphical user interface (GUI) component selectable by a user to access an associated function to generate an integrated circuit design”, as recited in claim 19. As discussed above, Nahaboo discloses that the interface development tool is separate from the application with which the interface is used, and that the developed interface is stored as a file with the application, which has an embedded WOOL processor to access and implement the interface file. *See Nahaboo*, col. 1, lines 35-39 and col. 2, lines 10-14 and 21-56. However, Nahaboo does not disclose or suggest that that user of the application (as opposed to the interface design tool) in any way specifies the interface. Rather, Nahaboo discloses that the interface designer uses a development tool to specify the interface. *See Nahaboo*, col. 1, lines 35-39. Accordingly, Nahaboo does not disclose or suggest “assembling a graphical user interface having no hard coded objects and including at least one graphical user interface (GUI) component based on interpreted configuration commands from the user, the at least one graphical user interface (GUI) component selectable by a user to access an associated function to generate an integrated circuit design”, as recited by claim 19.

Dangelo fails to overcome the deficiencies of Nahaboo. While Dangelo discloses that circuit specifications can be reused, Dangelo does not disclose or suggest “assembling a graphical user interface having no hard coded objects and including at least one graphical user interface (GUI) component based on interpreted configuration commands from the user, the at least one graphical user interface (GUI) component selectable by a user to access an associated function to generate an integrated circuit design”, as recited by claim 19.

Thus, the asserted combination of Nahaboo and Dangelo fails to disclose or suggest at least one element of claim 19, and of claims 20-23, at least by virtue of their dependency from allowable claim 19.

Further, the asserted combination of Nahaboo and Dangelo fails to disclose or suggest “the graphical user interface specified entirely by a user via a user input including one or more configuration commands provided to the command processor at run time of the command processor and interpreted by the command interpreter, wherein the configuration commands build graphical objects within the graphical user interface and assign functionality to the built graphical objects” and “one or more design tools corresponding to processes within an integrated circuit design process”, “wherein the one or more design tools operate under control of the command processor and within the graphical user interface”, as recited in claim 24. Instead, Nahaboo discloses that the interface designer uses an interface development tool to specify the interface, which tool is separate from the application with which the interface is operated. *See Nahaboo*, col. 1, lines 35-39, col. 2, lines 10-14 and 21-56. Dangelo discloses that circuit specifications can be reused, but fails to overcome the deficiencies of Nahaboo. The asserted combination of Nahaboo and Dangelo fails to disclose or suggest “the graphical user interface specified entirely by a user via a user input including one or more configuration commands provided to the command processor at run time of the command processor and interpreted by the command interpreter, wherein the configuration commands build graphical objects within the graphical user interface and assign functionality to the built graphical objects” and “wherein the one or more design tools operate under control of the command processor and within the graphical user interface”, as recited in claim 24, as recited in claim 19. Accordingly, the asserted combination of Nahaboo and Dangelo fails to disclose or suggest at least one element of claim 24, and of claims 25-28, at least by virtue of their dependency from allowable claim 24.

Further, the Applicant submits that the asserted combination of Nahaboo and Dangelo constitutes an impermissible hindsight reconstruction using the present application as a template to piece together elements from Nahaboo and Dangelo. The fact finder must be aware of distortion caused by hindsight bias and must be cautious of arguments reliant upon ex post reasoning. *See*

KSR Int'l Co. v. Teleflex Inc., citing *Graham v. John Deere*, 383 U.S., at 36. **“Determination of obviousness cannot be based on the hindsight combination of components selectively culled from the prior art to fit the parameters of the patented invention.”** *ATD Corp. v. Lydall, Inc.*, 159 F.3d 534, 48 USPQ2d 1321 (Fed. Cir. 1998) (emphasis added); *see also KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. ____ (2007), citing *Monroe Auto Equipment Co. v. Heckethorn Mfg & Supply Co.*, 332 F.2d 406, 412 (CA6 1964)) (warning against a “temptation to read into the prior art the teachings of the invention in issue”). This impermissible hindsight reconstruction is present here, where the Office ignores the explicit teachings of Nahaboo relating to separate applications including 1) an interface development tool to construct user interfaces and 2) an application with an embedded WOOL interpreter program to utilize the created interfaces to produce an interface for use with the application. The WOOL interpreter program is not indicated to allow a user to specify or control the interface and Nahaboo discloses that the interface development tool is used to modify the interface. The interface development tool is indicated to be usable with a variety of applications, specifically because the WOOL interpreter program is embedded with each of the applications to load the interface file. There is no teaching or suggestion to combine the interface development tool with the application. Notwithstanding the suggestion of the Office that the design application of Dangelo could be combined with the interface development tool, the teachings of Nahaboo teach away from such a combination. The only motivation for modifying either Nahaboo or Dangelo to provide the elements disclosed in the claims is provided by the present application.

Further, the Office is reminded that rejections on obviousness cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness. *See KSR Int'l Co. v. Teleflex Inc.*, citing *In re Kahn*, 441 F.3d 977, 988 (CA Fed. 2006). The Office suggests that because Nahaboo discloses “an extremely flexible development tool that can be used regardless of the application (column 1, lines 29-34)”, then “Dangelo would thus look to Nahaboo regarding features of modifying a graphical user interface to produce a design.” *See the Final Office Action*, p. 5, paragraph 5. Further, the Office asserts that Dangelo discloses that the interface of Dangelo may be specified for different users. *See the Final Office Action*, p. 5, paragraph 5. However, the Office

misconstrues the teaching of Dangelo, which discloses a design compiler that enables the user to follow the process of the design from one environment to another via script shells and command line and which discloses a GUI interface that facilitates user interactions by abstracting out those steps in the design flow that down require the designer's intervention. *See Dangelo*, p.col. 16, lines 16-42. Dangelo does not disclose or suggest a user specified GUI. Further, while Nahaboo recognizes a problem with poor interface design, Nahaboo does not disclose or suggest the interface design tool be combined with the application. Rather, Nahaboo specifically discloses that the created interface is stored in a file with the application and that an interpreter is embedded with the application to read and implement the interface file. *See Nahaboo*, col. 1, lines 35-39, col. 2, lines 10-14 and 21-56. Neither Nahaboo nor Dangelo provide any suggestion or motivation to incorporate an interface design tool with an application that generates integrated circuit designs.

Further, Nahaboo provide no suggestion or motivation for incorporating its interface design tool into the application, but rather teaches away from such incorporation by teaching embedding of a WOOL interpreter program instead. Accordingly, there exists no motivation for making the asserted combination. Therefore, the asserted combination of Nahaboo and Dangelo constitutes an impermissible hindsight reconstruction, and the rejection of the pending claims over the asserted combination of Nahaboo and Dangelo should be withdrawn.

CONCLUSION

With this response, the Applicant has identified specific elements of the claims that are not disclosed or suggested by the asserted combination of Nahaboo and Dangelo. Further, applicants have explained that the asserted combination constitutes an impermissible hindsight reconstruction, which should be withdrawn. With this response, all of the pending claims are in condition for allowance. Reconsideration and notice to that effect is respectfully requested.

The Examiner is invited to contact the undersigned attorney at the telephone number listed below if such a call would in any way facilitate allowance of this application. The Director is authorized to charge any fee deficiency required by this paper or credit any overpayment to Deposit Account No. 23-1123.

Respectfully submitted,

WESTMAN, CHAMPLIN & KELLY, P.A.

By: /R. Michael Reed/
R. Michael Reed, Reg. No. 59,066
900 Second Avenue South, Suite 1400
Minneapolis, Minnesota 55402-3319
Phone: (612) 334-3222 Fax: (612) 334-3312

RMR:rkp